

[Sign in](#)

Google

[Web](#) [Images](#) [Video](#) [News](#) [Maps](#) [more »](#)[Advanced Search](#)
[Preferences](#)**Web**Results 1 - 10 of about 38,500,000 for **state client web application**. (0.24 seconds)**[Editorial Draft] State in Web application design**

The **Web** browser is one half of the design of **state** in a **Web** application, ... The session **state** may be realized by storing **state** on the **client** or on the ...

www.w3.org/2001/tag/doc/state.html - 64k - [Cached](#) - [Similar pages](#)

Archive of W3C News in 2005

2005-11-15: W3C is pleased to announce the launch of the Rich **Web** **Client** Activity for **client-side** **Web** **Application** development. The **Web** APIs Working Group ...

www.w3.org/News/2005 - 193k - [Cached](#) - [Similar pages](#)

[[More results from www.w3.org](#)]

Designing Enterprise Applications with the J2EE Platform, Second ...

Web applications may support only one type of **client** with a single ... The section "Maintaining **Client** **State**" in The J2EE Tutorial explains in detail how to ...

java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/web-tier/web-tier5.html - 80k - [Cached](#) - [Similar pages](#)

Introduction to Web Forms State Management

View **state**, control **state**, hidden fields, cookies, and query strings all involve storing data on the **client** in various ways. However, **application** **state** ...

msdn2.microsoft.com/en-us/75x4ha6s.aspx - 27k - [Cached](#) - [Similar pages](#)

Microsoft ASP.NET QuickStarts Tutorial

Using **Application** **State** Using Session **State** Using **Client-Side** Cookies ... **Web** server (enter iisreset on the command line) and the session **state** value will ...

samples.gotdotnet.com/quickstart/aspplus/doc/stateoverview.aspx - 27k -

[Cached](#) - [Similar pages](#)

Web Client Software Factory Session State StateValue<T> for Unit ...

Web **Client** Software Factory Session **State** StateValue for Unit Testing - Patterns and ...

When unit testing **Application** Controller Classes, for example, ...

[www.davidhayden.com/blog/dave/archive/2007/](http://www.davidhayden.com/blog/dave/archive/2007/02/24/WebClientSoftwareFactorySessionStateStateValueT.aspx)

[02/24/WebClientSoftwareFactorySessionStateStateValueT.aspx](http://www.davidhayden.com/blog/dave/archive/2007/02/24/WebClientSoftwareFactorySessionStateStateValueT.aspx) - 48k -

[Cached](#) - [Similar pages](#)

Preparation Guide for Exam 70-528: TS: Microsoft .NET Framework ...

NET Framework 2.0 - **Web-based** **Client** Development. ... Manage **state** of an **application** by using **client-based** **state** management options. ...

www.microsoft.com/learning/exams/70-528.mspx - 76k - Mar 25, 2007 -

[Cached](#) - [Similar pages](#)

Client Side State - HTTP Cookies

The addition of a simple, persistent, **client-side** **state** significantly extends the capabilities of **Web-based** **client/server** applications. ...

wp.netscape.com/newsref/std/cookie_spec.html - 17k - [Cached](#) - [Similar pages](#)

Building Web Services the REST Way

The representation places the **client** **application** in a **state**. ... The **Web** is a REST system!

Many of those **Web** services that you have been using these many ...

www.xfront.com/REST-Web-Services.html - 18k - [Cached](#) - [Similar pages](#)

How To Handle Web Client Session State in an Application Center ...
Application Center 2000 does not replicate client state, including Active Server Pages (ASP) session variables, between cluster members.
support.microsoft.com/kb/286279 - [Similar pages](#)

Result Page: 1 [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) **[Next](#)**

Download [Google Pack](#): free essential software for your PC

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2007 Google

The Java EE 5 Tutorial

Home | Download | PDF | API | FAQ | Search | Feedback | History



Maintaining Client State

Many applications require that a series of requests from a client be associated with one another. For example, the Duke's Bookstore application saves the state of a user's shopping cart across requests. Web-based applications are responsible for maintaining such state, called a *session*, because HTTP is stateless. To support applications that need to maintain state, Java servlet technology provides an API for managing sessions and allows several mechanisms for implementing sessions.

Accessing a Session

Sessions are represented by an `HttpSession` object. You access a session by calling the `getSession` method of a request object. This method returns the current session associated with this request, or, if the request does not have a session, it creates one.

Associating Objects with a Session

You can associate object-valued attributes with a session by name. Such attributes are accessible by any web component that belongs to the same web context *and* is handling a request that is part of the same session.

The Duke's Bookstore application stores a customer's shopping cart as a session attribute. This allows the shopping cart to be saved between requests and also allows cooperating servlets to access the cart. `CatalogServlet` adds items to the cart; `ShowCartServlet` displays, deletes items from, and clears the cart; and `CashierServlet` retrieves the total cost of the books in the cart.

```
public class CashierServlet extends HttpServlet {
    public void doGet (HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        // Get the user's session and shopping cart
        HttpSession session = request.getSession();
        ShoppingCart cart =
            (ShoppingCart)session.
                getAttribute("cart");

        ...
        // Determine the total price of the user's books
        double total = cart.getTotal();
    }
}
```

Notifying Objects That Are Associated with a Session

Recall that your application can notify web context and session listener objects of servlet life-cycle events ([Handling Servlet Life-Cycle Events](#)). You can also notify objects of certain events related to their association with a session such as the following:

- When the object is added to or removed from a session. To receive this notification, your object must implement the `javax.servlet.http.HttpSessionBindingListener` interface.

- When the session to which the object is attached will be passivated or activated. A session will be passivated or activated when it is moved between virtual machines or saved to and restored from persistent storage. To receive this notification, your object must implement the `javax.servlet.http.HttpSessionActivationListener` interface.

Session Management

Because there is no way for an HTTP client to signal that it no longer needs a session, each session has an associated timeout so that its resources can be reclaimed. The timeout period can be accessed by using a session's `[get|set]MaxInactiveInterval` methods. You can also set the timeout period in the deployment descriptor using NetBeans 5.5:

1. Open the `web.xml` file in the `web.xml` editor.
2. Click General at the top of the editor.
3. Enter an integer value in the Session Timeout field. The integer value represents the number of minutes of inactivity that must pass before the session times out.

To ensure that an active session is not timed out, you should periodically access the session via service methods because this resets the session's time-to-live counter.

When a particular client interaction is finished, you use the session's `invalidate` method to invalidate a session on the server side and remove any session data. The bookstore application's `ReceiptServlet` is the last servlet to access a client's session, so it has the responsibility to invalidate the session:

```
public class ReceiptServlet extends HttpServlet {
    public void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        // Get the user's session and shopping cart
        HttpSession session = request.getSession();
        // Payment received -- invalidate the session
        session.invalidate();
        ...
    }
}
```

Session Tracking

A web container can use several methods to associate a session with a user, all of which involve passing an identifier between the client and the server. The identifier can be maintained on the client as a cookie, or the web component can include the identifier in every URL that is returned to the client.

If your application uses session objects, you must ensure that session tracking is enabled by having the application rewrite URLs whenever the client turns off cookies. You do this by calling the response's `encodeURL(URL)` method on all URLs returned by a servlet. This method includes the session ID in the URL only if cookies are disabled; otherwise, it returns the URL unchanged.

The `doGet` method of `ShowCartServlet` encodes the three URLs at the bottom of the shopping cart display page as follows:

```
out.println("<p> &nbsp; <p><strong><a href=\"" +
    response.encodeURL(request.getContextPath() +
        "/bookcatalog") +
    "\">" + messages.getString("ContinueShopping") +
    "</a> &nbsp; &nbsp; &nbsp; <a href=\"" +
    response.encodeURL(request.getContextPath() +
        "/bookcashier") +
    "\">" + messages.getString("Checkout") +
```

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	1	"20040205782"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 16:58
S2	552	717/116.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 14:14
S3	1468	719/315,316.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 14:17
S4	19945	application with class	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 14:17
S5	552	717/116.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 14:33
S6	18	web near2 application same class same navigat\$4	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 14:37
S7	705	web near2 application same class	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 14:38
S8	0	web near2 application same class same lauch\$4	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 14:38
S9	46	web near2 application same class same (start\$ shut)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 14:39
S10	4107	state same web near page\$1	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 16:59
S11	552	717/116.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 16:59

EAST Search History

S12	1468	719/315,316.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 16:59
S13	24	S10 and (S11 S12)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 17:03
S14	26	web and client same persist near5 state	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 11:10
S15	1	(S11 S12) and S14	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 17:04
S16	112	web near2 page\$1 and persist near5 state	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 17:10
S17	1	(S11 S12) and S16	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 17:09
S18	35	web and page\$1 same persist near5 state	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 17:13
S19	11058	web and page\$1 same state	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/26 17:14
S20	1858	client same cookies	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 11:07
S21	512	client same cookies same state	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 11:07
S22	7067	719/3\$.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 11:08
S23	41956	"707"/\$.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 11:08

EAST Search History

S24	552	717/116.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 11:09
S25	46	client same persist near5 state	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 11:11
S26	6	S20 and S25	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 11:11
S27	512	client same cookies same state	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 13:42
S28	7067	719/3\$.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 13:40
S29	552	717/116.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 13:40
S30	41956	"707"/\$.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 13:40
S31	88	S27 and (S28 S29 S30)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 13:41
S32	269	client same cookies same state same store\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 13:42
S33	48	S32 and (S28 S29 S30)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 13:44
S34	89601	page\$1 same display\$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 13:44
S35	35	S33 and S34	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2007/03/27 13:45